

# ĐỀ THI PYTHON CƠ BẢN

Thời gian: 120 phút Ngày thi: 13/11/2021 - Được sử dụng tài liệu.

Khi nộp bài, học viên gởi các file bài làm của mình (chỉ gởi các file .py) đến email levanhanhyn@gmail.com.

# <u>CÂU 1.</u> (4 điểm) Lần lượt cài đặt các hàm có trong yêu cầu, sau đó viết chương trình gọi các hàm này thực hiện:

- a. (1đ) Viết hàm cho người dùng nhập số nguyên dương n có giá trị nằm trong khoảng từ 10 đến 1000. Nếu nhập đúng, hàm tạo ra 1 list chứa n các giá trị ngẫu nhiên không trùng nhau nằm trong khoảng từ 1 đến 5000 và hàm trả về list vừa tạo được. Ngược lại nếu nhập sai, chương trình sẽ yêu cầu nhập lại cho đến khi nhập đúng. Các trường hợp được xem là n nhập sai:
  - (i)- Nếu người dùng nhập kiểu dữ liệu không phải là số nguyên (như string, float, bool, ...) chương trình sẽ hiện ra câu báo lỗi '*Phải nhập số nguyên dương*. Yêu cầu nhập lại'.
  - (ii)- Nếu giá trị nhập vào đúng kiểu dữ liệu nguyên nhưng giá trị không nằm trong khoảng 10 đến 1000, chương trình báo lỗi '*Chỉ nhận giá trị từ 10 đến 1000. Yêu cầu nhập lại*'.

<u>Yêu cầu cài đặt</u>: phải sử dụng lệnh try ... except để xử lý các exception có thể xảy ra trong chương trình.

## b. (1đ) Số thân thiện:

Số tự nhiên có rất nhiều tính chất thú vị. Ví dụ với số 23, số đảo ngược của nó là 32. Hai số này có ước chung lớn nhất là 1. Những số như thế được gọi là số thân thiện, tức là số 23 được gọi là số thân thiên, số 32 cũng được gọi là số thân thiên.

Các số thân thiện có giá trị nhỏ hơn 100 gồm: 1, 10, 13, 14, 16, 17, 19, 23, 25, 29, 31, 32, 34, 35, 37, 38, 41, 43, 47, 49, 52, 53, 56, 58, 59, 61, 65, 67, 71, 73, 74, 76, 79, 83, 85, 89, 91, 92, 94, 95, 97, 98.

- E <u>Yêu cầu cài đặt</u>: Viết hàm liệt kê các số thân thiện có trong list vừa tạo ở câu a.
- E Kết quả gọi ý khi thực hiện chương trình:
  - Nếu trong list không có số thân thiện nào thì in ra thông báo:

Trong list không chứa số thân thiện.

• Nếu trong list có số thân thiên nào thì in ra thông báo:

Các số thân thiên có trong list là ...

#### c. (1đ) Số nguyên tố strobogrammatic

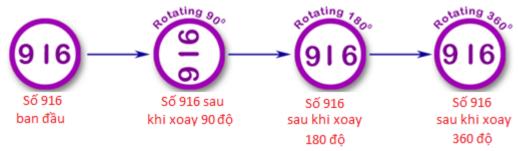
► Giả sử có định nghĩa về số strobogrammatic và số nguyên tố strobogrammatic như sau:

#### ☐ Số strobogrammatic

- Là một số có giá trị không đổi khi xoay số đó 180 độ (180°). Nhận xét: các số strobogrammatic chỉ chứa các số sau đây: 0, 1, 6, 8, 9. Trong đó các số 0, 1, 8 không bị thay đổi giá trị sau khi xoay, còn số 6 và 9 bị thay đổi (6 chuyển thành 9 và 9 chuyển thành 6).
- Ví dụ 1: số 68910 sau khi xoay sẽ là 01689 => 68910 không phải là số

Strobogrammatic vì 68910 ≠ 01689

• Ví du 2:



Minh họa các số strobogrammatic đầu tiên là 0, 1, 8, 11, 69, 88, 96, 101, 111, 181, 609, 619, 689, 808, 818, 888, 906, 916, 986, 1001, 1111, 1691, 1881, 1961, 6009, 6119, 6699, 6889, 6969, 8008, 8118, 8698, 8888, 8968, 9006, 9116, 9696, 9886, 9966, ...

## Số nguyên tố strobogrammatic

- Là số vừa là số nguyên tố, vừa là số strobogrammatic.
- Minh họa các số nguyên tố *strobogrammatic* đầu tiên là: 11, 101, 181, 619, 16091, 18181, 19861, 61819, 116911, 119611, 160091, 169691, 191161, 196961, 686989, 688889, ...
- Yêu cầu cài đặt: Viết hàm liệt kê các số nguyên tố strobogrammatic có trong list vừa tạo ở câu a.
- E <u>Kết quả gợi ý khi thực hiện chương trình</u>: tương tự gợi ý khi thực hiện chương trình ở câu b.

## d. (1đ) Lucky Number

#### ₽ Lucky Number

' (Theo Wikipedia) Lucky Number (số may mắn) là số được định nghĩa theo quá trình sau: bắt đầu với số nguyên dương x và tính tổng bình phương y các chữ số của x, sau đó tiếp tục tính tổng bình phương các chữ số của y. Quá trình này lặp đi lặp lại cho đến khi thu được kết quả là 1 thì dừng (tổng bình phương các chữ số của số 1 chính là 1) hoặc quá trình sẽ kéo dài vô tận. Số mà quá trình tính này kết thúc bằng 1 gọi là số may mắn. Số có quá trình tính kéo dài vô tận là số không may mắn hay còn gọi là sad number (số đen đủi)

Ví dụ: 7 là số may mắn vì  

$$7^2 = 49$$
  
 $4^2 + 9^2 = 97$   
 $9^2 + 7^2 = 130$   
 $1^2 + 3^2 + 0^2 = 10$   
 $1^2 + 0^2 = 1$ 

- Minh họa những số may mắn dưới 500 là: 1, 7, 10, 13, 19, 23, 28, 31, 32, 44, 49, 68, 70, 79, 82, 86, 91, 94, 97, 100, 103, 109, 129, 130, 133, 139, 167, 176, 188, 190, 192, 193, 203, 208, 219, 226, 230, 236, 239, 262, 263, 280, 291, 293, 301, 302, 310, 313, 319, 320, 326, 329, 331, 338, 356, 362, 365, 367, 368, 376, 379, 383, 386, 391, 392, 397, 404, 409, 440, 446, 464, 469, 478, 487, 490, 496.
- E Yêu cầu cài đặt: Viết hàm liệt kê các số may mắn có trong list vừa tạo ở câu a.

Giả sử để xác định quá trình tìm *Lucky number* có kéo dài vô tận hay không, người ta cho lặp việc tìm này tối đa 100 lần. Do đó nếu quá trình lặp để tìm vượt trên 100 lần thì chương trình có thể kết luận số đó không là *Lucky number*.

E <u>Kết quả gơi ý khi thực hiện chương trình</u>: tương tự gợi ý khi thực hiện chương trình ở câu b.

# TRUNG TÂM TIN HỌC ĐẠI HỌC KHOA HỌC TỰ NHIÊN TP. HỒ CHÍ MINH



# <u>CÂU 2.</u> (2 điểm) Viết chương trình sử dụng kỹ thuật Comprehension để giải quyết các yêu cầu sau:

a. (1đ) Viết chương trình cho người dùng nhập chuỗi (S) là 1 địa chỉ IP (IP Address). Thực hiện xóa các số 0 (zero) không có nghĩa.

Ví dụ: 
$$S=$$
 '192.024.001.023'  $\Rightarrow$  sẽ in ra 192.24.1.23  
Hoặc  $S=$  '005.004.001.020'  $\Rightarrow$  sẽ in ra 5.4.1.20  
Cho nhập số nguyên dương  $n$ . In ra các số nguyên dương  $X$  nhỏ hơn hay bằng  $n$ , sao cho các ký số có trong  $X$  đều là số chẵn.

b. (1đ) Bài toán gà chó. Nội dung bài toán như sau:

Vừa gà vừa chó Bó lại cho tròn Đúng ba sáu con Một trăm chân chẵn.

Viết chương trình tìm tất cả các nghiệm có thể có của bài toán. Trong mỗi trường hợp của kết quả, cho biết số lượng gà, số lượng chó?

## <u>CÂU 3.</u> (1 điểm)

- Yêu cầu: Xây dựng hàm ẩn danh (anonynous function) cho các trường hợp sau với yêu cầu trong thân hàm ẩn danh không được gọi hàm bên ngoài hay nói cách khác là bên trong thân hàm phải giải quyết được yêu cầu của đề bài
  - a.- (0.5đ) Xây dựng hàm ẩn danh nhận 2 tham số là a và b, hàm thực hiện giải phương trình bậc 1: ax + b = 0.
  - b.- (0.5đ) Xây dựng hàm ẩn danh nhận 6 số thực a, b, c, d, e, f. Hàm thực hiện giải hệ phương trình sau:

$$\begin{cases} ax + by = c \\ dx + ey = f \end{cases}$$

Nnac lại: tinh

$$D = \begin{vmatrix} a & b & = ae - \\ d & e & bd \end{vmatrix}; \quad Dx = \begin{vmatrix} c & b \\ f & e \end{vmatrix} = ce - bf ; \quad Dy = \begin{vmatrix} a & c \\ d & f \end{vmatrix} = af - dc$$

$$N = \begin{vmatrix} c & b \\ f & e \end{vmatrix} = ce - bf ; \quad Dy = \begin{vmatrix} a & c \\ d & f \end{vmatrix} = af - dc$$

$$N = \begin{vmatrix} c & b \\ f & e \end{vmatrix} = ce - bf ; \quad Dy = \begin{vmatrix} a & c \\ d & f \end{vmatrix} = af - dc$$

$$N = \begin{vmatrix} c & b \\ f & e \end{vmatrix} = ce - bf ; \quad Dy = \begin{vmatrix} a & c \\ d & f \end{vmatrix} = af - dc$$

$$N = \begin{vmatrix} c & b \\ f & e \end{vmatrix} = ce - bf ; \quad Dy = \begin{vmatrix} a & c \\ d & f \end{vmatrix} = af - dc$$

$$N = \begin{vmatrix} c & b \\ f & e \end{vmatrix} = ce - bf ; \quad Dy = \begin{vmatrix} a & c \\ d & f \end{vmatrix} = af - dc$$

$$N = \begin{vmatrix} c & b \\ f & e \end{vmatrix} = ce - bf ; \quad Dy = \begin{vmatrix} a & c \\ d & f \end{vmatrix} = af - dc$$

$$N = \begin{vmatrix} c & b \\ f & e \end{vmatrix} = af - dc$$

$$N = \begin{vmatrix} c & b \\ f & e \end{vmatrix} = af - dc$$

$$N = \begin{vmatrix} c & b \\ f & e \end{vmatrix} = af - dc$$

$$N = \begin{vmatrix} c & b \\ f & e \end{vmatrix} = af - dc$$

$$N = \begin{vmatrix} c & b \\ f & e \end{vmatrix} = af - dc$$

$$N = \begin{vmatrix} c & b \\ f & e \end{vmatrix} = af - dc$$

$$N = \begin{vmatrix} c & b \\ f & e \end{vmatrix} = af - dc$$

$$N = \begin{vmatrix} c & b \\ f & e \end{vmatrix} = af - dc$$

$$N = \begin{vmatrix} c & b \\ f & e \end{vmatrix} = af - dc$$

$$N = \begin{vmatrix} c & b \\ f & e \end{vmatrix} = af - dc$$

$$N = \begin{vmatrix} c & b \\ f & e \end{vmatrix} = af - dc$$

$$N = \begin{vmatrix} c & b \\ f & e \end{vmatrix} = af - dc$$

$$N = \begin{vmatrix} c & b \\ f & e \end{vmatrix} = af - dc$$

$$N = \begin{vmatrix} c & b \\ f & e \end{vmatrix} = af - dc$$

$$N = \begin{vmatrix} c & b \\ f & e \end{vmatrix} = af - dc$$

$$N = \begin{vmatrix} c & b \\ f & e \end{vmatrix} = af - dc$$

$$N = \begin{vmatrix} c & b \\ f & e \end{vmatrix} = af - dc$$

$$N = \begin{vmatrix} c & b \\ f & e \end{vmatrix} = af - dc$$

$$N = \begin{vmatrix} c & b \\ f & e \end{vmatrix} = af - dc$$

$$N = \begin{vmatrix} c & b \\ f & e \end{vmatrix} = af - dc$$

$$N = \begin{vmatrix} c & b \\ f & e \end{vmatrix} = af - dc$$

$$N = \begin{vmatrix} c & b \\ f & e \end{vmatrix} = af - dc$$

$$N = af - c$$

$$N$$

# CÂU 4. (3 điểm) Viết chương trình đọc và ghi file CSV

Đầu giờ thi, GV sẽ cung cấp file *WHO-COVID-19-global-table-data.csv* để HV làm bài. Nội dung file này thống kê số lượng bệnh nhân COVID-19, số tử vong ở các nước trong thời gian qua, với các thông tin cụ thể được mô tả trong bảng sau:

Tên cột	Kiểu dữ liệu	Mô tả
Name	String	Quốc gia, lãnh thổ, khu vực
WHO_region	String	Khu vực của WHO
Cases - cumulative total	Integer	Các trường hợp được xác nhận tích lũy được báo cáo cho WHO cho đến nay.
Cases - cumulative total per 100000 population	Decimal	Các trường hợp được xác nhận tích lũy được báo cáo cho WHO cho đến nay trên 100.000 dân.
Cases - newly reported in last 7 days	Integer	Các trường hợp được xác nhận mới được báo cáo trong 7 ngày qua. Được tính bằng cách trừ số trường hợp tích lũy trước đó (8 ngày trước) cho số trường hợp tích lũy hiện tại.



## TRUNG TÂM TIN HỌC ĐẠI HỌC KHOA HỌC TỰ NHIÊN TP. HỒ CHÍ MINH

	1		
Cases - newly reported in last		Các trường hợp được xác nhận mới được báo cáo trong 7	
7 days per 100000 population	Decimal	ngày qua trên 100.000 dân.	
Cases - newly reported in last		Các trường hợp được xác nhận mới được báo cáo trong 24 giờ	
24 hours		qua. Được tính bằng cách trừ số trường hợp tích lũy trước đó	
	Integer	cho số trường hợp tích lũy hiện tại.	
Deaths - cumulative total		Các trường hợp tử vong được xác nhận tích lũy đã được báo	
	Integer	cáo cho WHO cho đến nay.	
Deaths - cumulative total per		Tính đến nay, số ca tử vong đã được xác nhận tích lũy được	
100000 population	Decimal	báo cáo cho WHO trên 100.000 dân.	
Deaths - newly reported in last		Các trường hợp tử vong mới được xác nhận được báo cáo	
7 days		trong 7 ngày qua. Được tính bằng cách trừ số tử vong tích lũy	
	Integer	trước đó (8 ngày trước) cho số tử vong tích lũy hiện tại.	
Deaths - newly reported in last			
7 days per 100000 population		Các trường hợp tử vong mới được xác nhận được báo cáo	
	Decimal	trong 7 ngày qua trên 100.000 dân.	
Deaths - newly reported in last		Các trường hợp tử vong mới được xác nhận được báo cáo	
24 hours		trong 24 giờ qua. Được tính bằng cách trừ số tử vong tích lũy	
	Integer	trước đó cho số tử vong tích lũy hiện tại.	

## ₽ Yêu cầu thực hiên:

- a. (1đ) Viết hàm InNoiDungFile\_CSV(filename) để đọc và in nội dung tập tin csv (được truyền vào thông qua tham số filename) ra màn hình theo mẫu sau biết rằng các nước có trong bảng kết quả là những nước thỏa cả 3 điểu kiện sau:
  - WHO Region là "Western Pacific"
  - Cases cumulative total: có giá trị > 50000
  - Cases newly reported in last 24 hours: giá trị nằm trong khoảng từ 100 đến 2500

#### WESTERN PACIFIC REGION

Name	Cases - cumulative total	Cases - newly reported in last 24 hours
Republic of Korea	243317	1882
Mongolia	199274	2433
Cambodia	90535	428
Singapore	66812	120
SUM	599938	4863

- b. (1đ) Viết hàm DocFile\_CSV(filename) để chọn ra những quốc gia có vùng lãnh thổ (WHO Region) thuộc 1 trong các vùng lãnh thổ: Africa, Europe, South-East Asia và từ trước đến nay chưa phát hiện bất kỳ ca nhiễm COVID-19 nào. Hàm trả về một list gồm 2 thông tin: tên quốc gia (Name) và vùng lãnh thổ (WHO Region).
- c. (1đ) Viết hàm *GhiFile\_CSV* (*filename*, *listname*) để ghi nội dung list lst vừa có ở câu b vào vào tập tin ketqua.csv (được khai báo thông qua filename).

Nội dung file kết quả (ketqua.csv) như sau:

Name,WHO Region Democratic People's Republic of Korea,South-East Asia Saint Helena,Africa Turkmenistan,Europe

## --- Chúc các ban làm bài tốt ---

Đề thi Python cơ bản K267 Trang 4/4